



CDP Support

Selenium 4 introduced a new API that utilizes CDP (Chrome DevTools Protocol) to allow the user to interact with Chrome DevTools on Chromium-based browsers.

 CDP is supported on Microsoft Edge and Chrome. Firefox only partially supports CDP API.

 CDP commands will not appear in the report

Examples

CDP allows the user to do several things with the Chrome DevTools, we will describe some examples:

- [Listen to console logs](#)
- [Change Geolocation](#)
- [Network interception](#)

Listen to console logs

In this example, we will print the console logs of the browser.

Console logs test

```
private WebDriver driver;

@BeforeEach
void setUp() throws MalformedURLException {
    DesiredCapabilities dc = new DesiredCapabilities();
    dc.setCapability(CapabilityType.BROWSER_NAME, Browser.CHROME.browserName());
    dc.setCapability("accessKey", <ACCESS_KEY>);

    driver = new RemoteWebDriver(new URL(<CLOUD_URL>), dc);
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
}

@Test
void consoleLogsTest() {
    driver = new Augmenter().augment(driver);
    DevTools devTools = ((HasDevTools) driver).getDevTools();
    devTools.createSession();
    devTools.send(Log.enable());
    devTools.addListener(Log.entryAdded(),
        logEntry -> {
            System.out.println("log: "+logEntry.getText());
            System.out.println("level: "+logEntry.getLevel());
        });
    driver.get("http://the-internet.herokuapp.com/broken_images");
}

@AfterEach
public void tearDown() {
    System.out.println("Report URL: "+ ((HasCapabilities) driver).getCapabilities().getCapability
("reportUrl"));
    driver.quit();
}
```

Change Geolocation

In this example, we will change the geolocation of the browser.

Geolocation test

```
private WebDriver driver;

@BeforeEach
void setUp() throws MalformedURLException {
    DesiredCapabilities dc = new DesiredCapabilities();
    dc.setCapability(CapabilityType.BROWSER_NAME, Browser.CHROME.browserName());
    dc.setCapability("accessKey", <ACCESS_KEY>);

    driver = new RemoteWebDriver(new URL(<CLOUD_URL>), dc);
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
}

@Test
void geolocationTest() {
    double expectedLatitude = 52.5043;
    double expectedLongitude = 13.4501;

    driver = new Augmenter().augment(driver);
    driver.get("https://my-location.org/");

    // Change geolocation
    DevTools devTools = ((HasDevTools) driver).getDevTools();
    devTools.createSession();
    devTools.send(Emulation.setGeolocationOverride(Optional.of(expectedLatitude),
        Optional.of(expectedLongitude),
        Optional.of(1)));

    driver.get("https://my-location.org/");

    // Extract new geolocation
    double actualLatitude = Double.parseDouble(driver.findElement(By.xpath("//*[@id='latitude']").
getText());
    double actualLongitude = Double.parseDouble(driver.findElement(By.xpath("//*[@id='longitude']").
getText());

    Assertions.assertEquals(expectedLatitude, actualLatitude, "Wrong latitude");
    Assertions.assertEquals(expectedLongitude, actualLongitude, "Wrong longitude");
}

@AfterEach
public void tearDown() {
    System.out.println("Report URL: " + ((HasCapabilities) driver).getCapabilities().getCapability
("reportUrl"));
    driver.quit();
}
```

Network interception

In this example, we will intercept network communication with the browser to show a message that we want.

Network interception test

```
private WebDriver driver;

@BeforeEach
void setUp() throws MalformedURLException {
    DesiredCapabilities dc = new DesiredCapabilities();
    dc.setCapability(CapabilityType.BROWSER_NAME, Browser.CHROME.browserName());
    dc.setCapability("accessKey", <ACCESS_KEY>);

    driver = new RemoteWebDriver(new URL(<CLOUD_URL>), dc);
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
}

@Test
void networkInterceptionTest() {
    driver = new Augmenter().augment(driver);
    NetworkInterceptor interceptor = new NetworkInterceptor(
        driver,
        Route.matching(req -> true)
            .to(() -> req -> new HttpResponse()
                .setStatus(200)
                .addHeader("Content-Type", MediaType.HTML_UTF_8.toString())
                .setContent(utf8String("Creamy, delicious cheese!"))));

    driver.get("https://example-sausages-site.com");

    String source = driver.getPageSource();
    Assertions.assertTrue(source.contains("delicious cheese!"));
}

@AfterEach
public void tearDown() {
    System.out.println("Report URL: " + ((HasCapabilities) driver).getCapabilities().getCapability(
        "reportUrl"));
    driver.quit();
}
```